







Historia da WEB



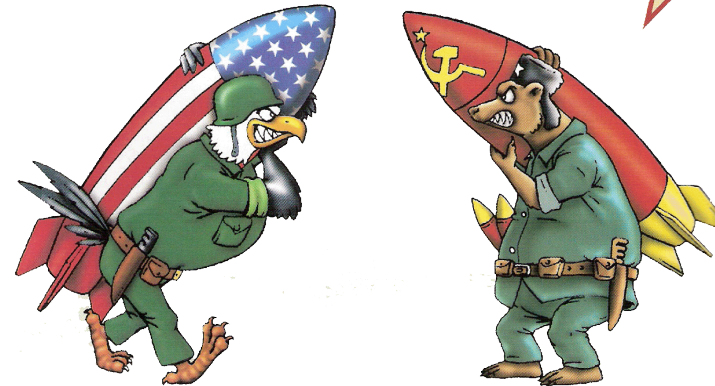
60-70

#Criação da ARPANET

Objetivo de ligar bases militares e os departamentos de Pesquisa do Governo Americano

- MIT – Instituto de Tecnologia de Massachusetts
- Criado pela DARPA - Agência de Projetos de Pesquisa Avançada
- Utilizava protocolo NCP

Como funciona: chaveamento de pacotes, que é um sistema de transmissão de dados em rede de computadores no qual as informações são divididas em pequenos pacotes, que por sua vez contém trecho dos dados, o endereço do destinatário e informações que permitiam a remontagem da mensagem original

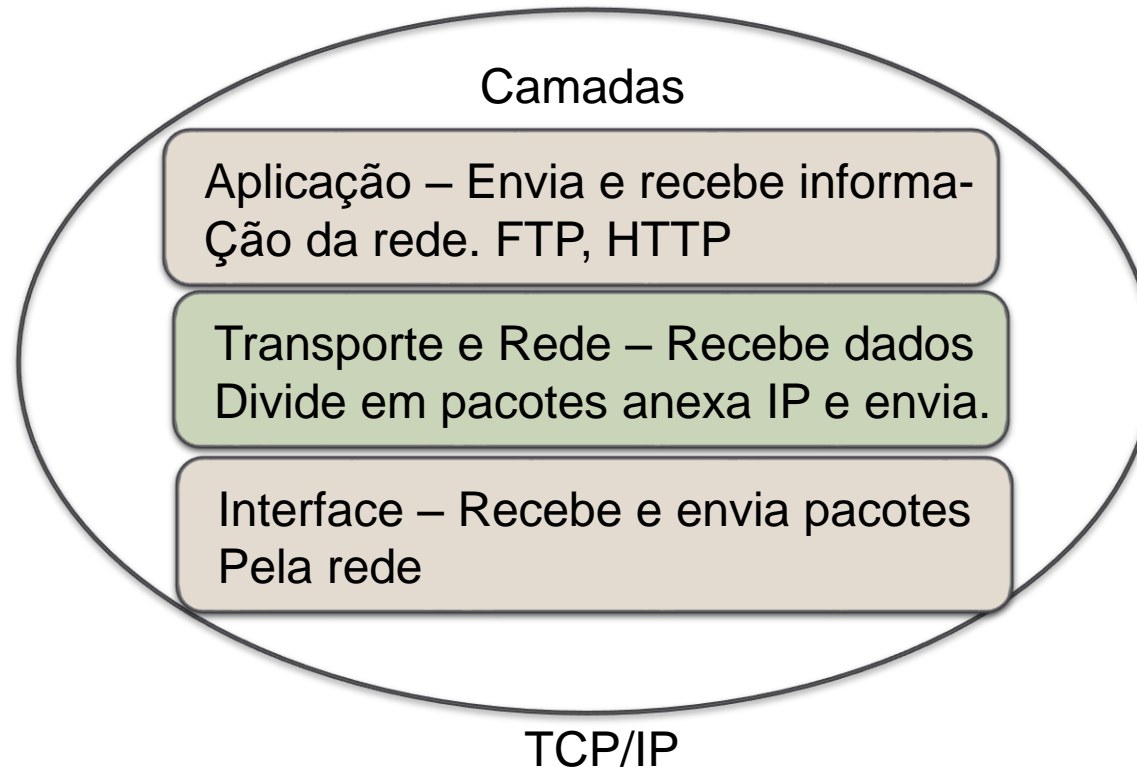
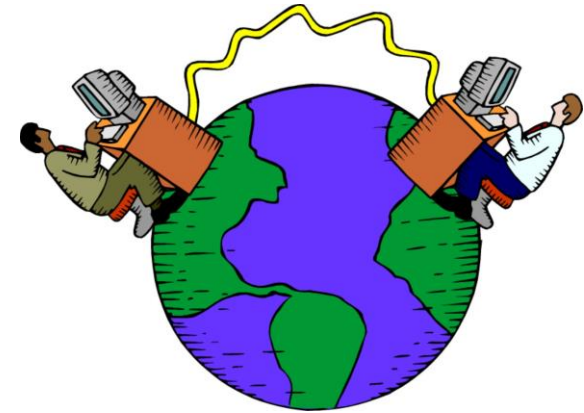




70-80

#Surge o termo Internet (internetworking)

- Nasce o Protocolo TCP/IP
- Criado pela ARPA e parceria com outras agências





80-90





#Popularização




- 1 de Janeiro de 83 – Flag Day – Arpanet adota TCP/IP como protocolo padrão
- Qualquer rede com IP poderia navegar pelos arquivos e trocar msg
- Nasce conjunto de redes universitárias interconectadas em 56Kbps intitulada – *National Science Foundation Network*
- EUA comercializa a ideia da internet e do hipertexto
- No Brasil 1988, projeto para ligar universidades daqui a instituições americanas
- Surge no Brasil Alternex o primeiro provedor de acesso a permiti pessoas físicas



90-00

 Tim Berners-Lee   

+Fábio 

Web      Ferramentas de pesquisa 

Aproximadamente 831.000 resultados (0,53 segundos)

Tim Berners-Lee – Wikipédia, a enciclopédia livre
pt.wikipedia.org/wiki/Tim_Berners-Lee ▼
Timothy John Berners-Lee KBE, OM, FRS (TimBL ou TBL) (Londres, 8 de junho de 1955) é um físico Foi saudado e chamado de herói por **Tim Berners-Lee**.
Vida e carreira - O primeiro sítio Web - Encontro virtual com Edward ... - Referências

Tim Berners-Lee - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Tim_Berners-Lee ▼ Traduzir esta página
Sir Timothy John "Tim" Berners-Lee, OM, KBE, FRS, FREng, FRSA, DFBCS (born 8 June 1955), also known as TimBL, is an English computer scientist, best ...

Tim Berners-Lee - World Wide Web Consortium
www.w3.org/People/Berners-Lee/ ▼ Traduzir esta página
Tim Berners-Lee invented the World Wide Web in 1989. He founded and Directs the World Wide Consortium (W3C) the forum for technical development of the ...

Tim Berners-Lee - Veja - Abril.com
veja.abril.com.br/especiais/tecnologia_2006/p_040.html ▼
VIDA DIGITAL ... e ele criou a web. O físico Tim Berners-Lee, o "pai da www", prevê novo salto da rede com intercâmbio maior de dados. Alessandro Greco ...

Tim Berners-Lee

 Mais imagens

Tim Berners-Lee

Cientista da computação

Timothy John Berners-Lee KBE, OM, FRS é um físico britânico, cientista da computação e professor do MIT. É o criador da World Wide Web, tendo feito a primeira proposta para sua criação a 25 de março de 1989. [Wikipédia](#)

Nascimento: 8 de junho de 1955 (59 anos), Londres, Reino Unido

Cônjuge: Rosemary Leith (desde 2014)

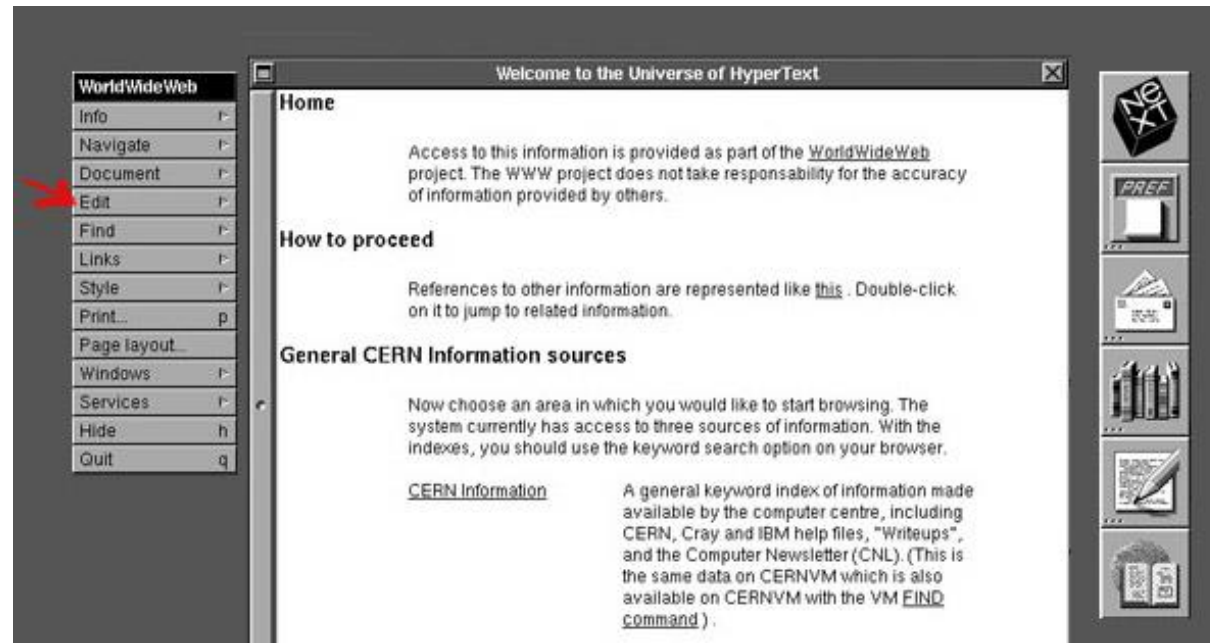
Obra: Weaving The Web



90-00

#World Wide Web

- 1992 - Tim Berners-Lee trabalhava na CERN(Organ. Europeia de Pesquisa Nuclear) foi aonde a Web foi inventada
- 1990 - Tim criou o primeiro prototipo de navegador a rodar em computadores da NeXT chamado Nexus
- Criação das noções de web, links e hiperlinks





90-00

#HTML – HyperText Markup Language

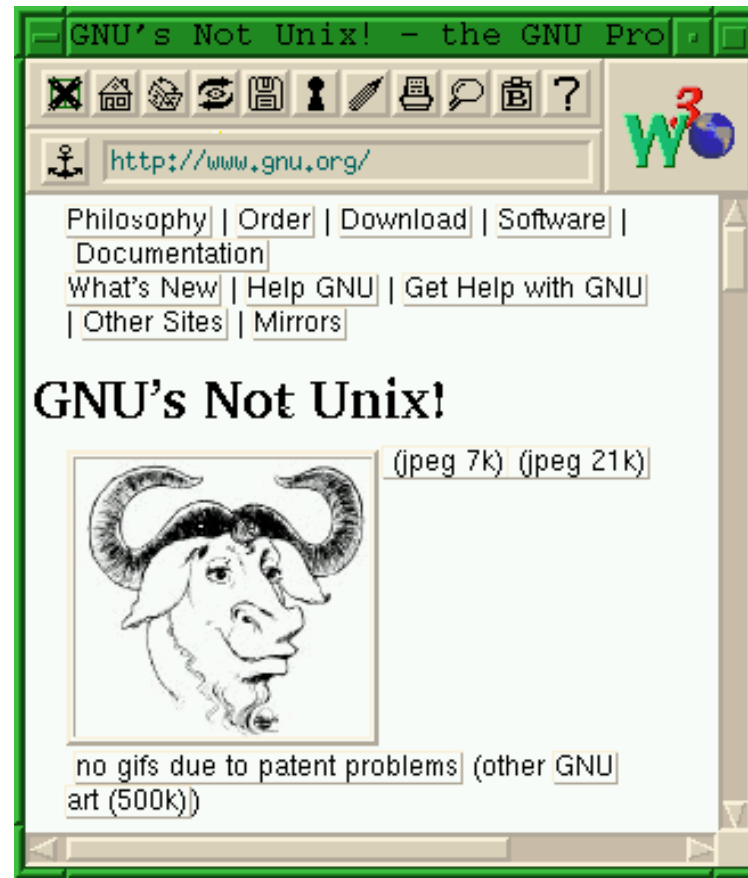
- 1990 - Tim desenvolveu um software e um protocolo para recuperar hipertextos chamado **HTTP**
- O formato de texto que criou para o HTTP foi chamado de **HTML** que usa a especificação do **SGML** (Normas Gerais para uso de linguagem de marcação)
- 1991 criada primeira lista de discussão eletrônica **WWW** talk – para trocar ideias sobre o HTML
- 1992 **NCSA** (Centro Nac. de Aplic. para Super Comput.) criam o navegador **MOSAIC**
- 1993 Dave Ragget (HP) publicou o HTML+ (Títulos, parágrafos, listas, tabelas e figuras)



90-00

#Browsers

- 1993 Dave Ragget com o HTML+ criou o navegador Arena

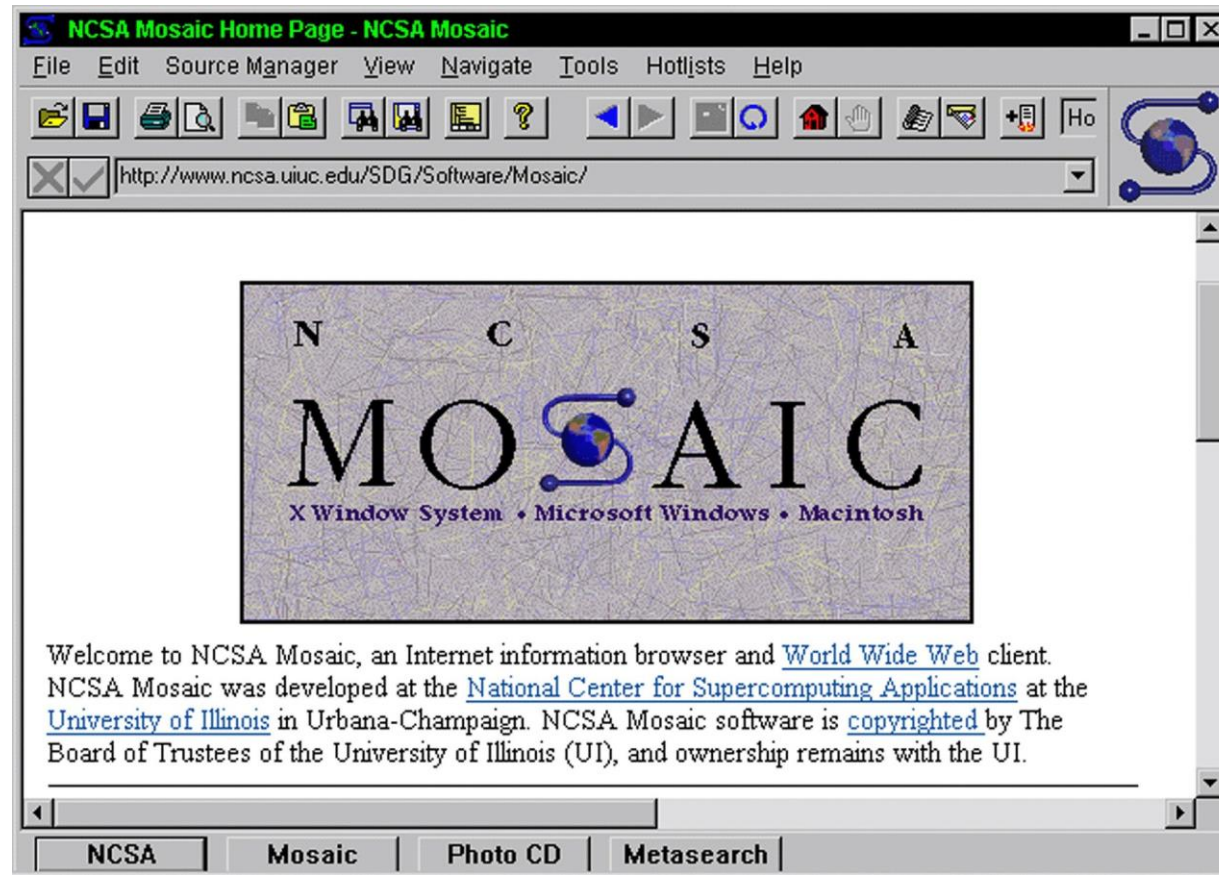




90-00

#Browsers

- 1993 Sun Microsystem lança a versão 1 do Mosaic





90-00

#HTML

- A web vira um caos com vários navegadores e versões de HTML
- 1994 Tim em parceria com a **CERN** criam a **W3C** (Word Wide Web Consortium) com a ideia de padronizar a web, em parceria com a MIT
- 1995 publicado o **HTML 2.0**
- Netscape implementa **frames** no HTML
- Microsoft com a versão 2 do I.E. implementa o **CSS**(folha de estilo em cascata)
- W3C cria o **HTML ERB** que vira o grupo de trabalho da HTML em parcerias com outras empresas



90-00

#HTML


- 01/1997 W3C publica **HTML3.2** como recomendação oficial
- 12/1997 W3C publica o **HTML4**
- 1999 W3C publica o **HTML4.01**





#HTML

- 2004 – **WEB 2.0** – Web como Plataforma - Serviço
- Popularização das redes sociais – uso do **AJAX**
- 2004 Criado a WHATWG – Grupo de trabalho para tecnologias de hipertexto, em conjunto da Apple, Mozilla e Opera. Com o intuito de propor para a W3C as especificações para o **HTML5**, Web forms 2.0 e Web controls 1.0
- WHATWG entra em parceria com a W3C para desenvolvimento do HTML5
- 2011 W3C fica dando continuidade ao desenvolvimento do HTML5
- 2012 W3C finaliza as especificações do HTML5



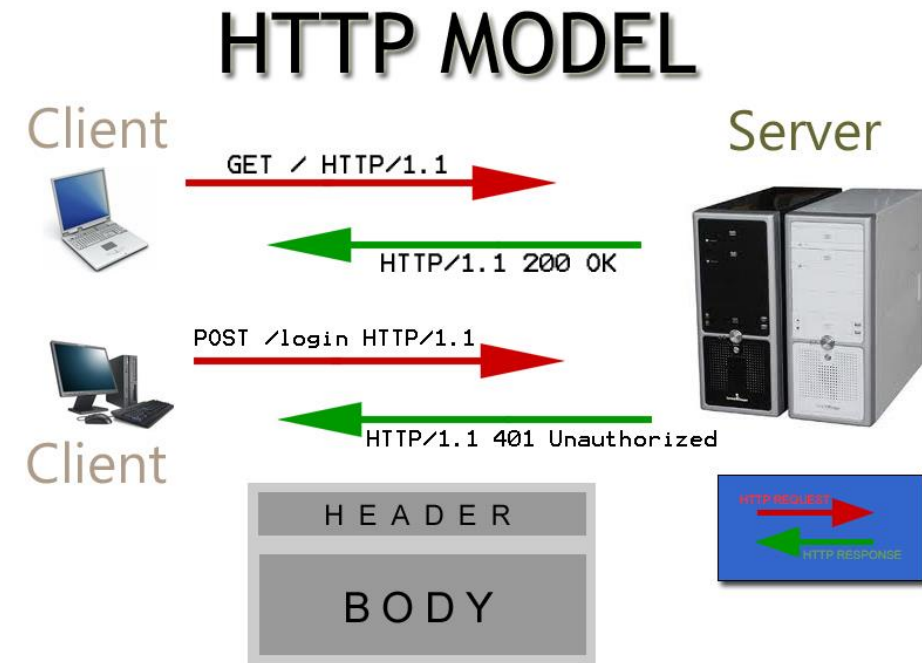
http://www

Hypertext Transfer Protocol



#Historia

- Protocolo de comunicação utilizado para enviar e receber dados na web
- Usa conexões TCP, tem por padrão a porta 80
- Versão atual: HTTP/1.1 - lançado em 1999
- Especificação RFC 2616
- É um protocolo sem estado (stateless)
- Para persistir informação usamos Cookies e Sessões
- HTTPS adiciona criptografia a comunicação com o servidor através de um protocolo seguro (TLS ou SSL)





#Historia

- A partir da Web 2.0 o HTTP/1.1 começou a se tornar um gargalo
- Google lançou o SPDY seu próprio protocolo
- SPDY funciona como uma camada a mais em cima do HTTP
- Todos os navegadores atuais e linguagens já suportam o SPDY
- Google e Facebook já trabalham em cima desse Protocolo
- Motivou a criação do HTTP/2
- HTTP/2 tem suporte a Multiplexação*
- HTTP/2 esta em fase de teste



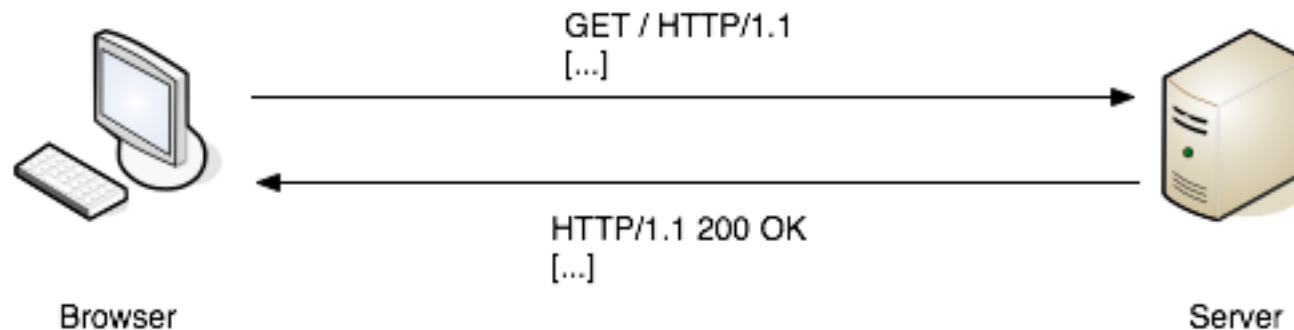
* Uma única conexão para baixar múltiplos arquivos



#Requisições

A requisição/resposta são dividido em três componentes:

Requisição	Resposta
Cabeçalho da mensagem	Status Protocolo
Corpo da mensagem	Cabeçalho de resposta
Requisição	Corpo da entidade





#Cabeçalho da mensagem

Cabeçalho de Solicitação de uma pagina feito pelo navegador(User agent)

Chave	Valor
Solicitar	GET /blog/ HTTP/1.1
Accept	text/html, application/xhtml+xml, */*
Accept-Language	pt-BR
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding	gzip, deflate
Host	www.fabioelisio.com
DNT	1
Connection	Keep-Alive
Cookie	__utma=138138919.1471094785.1425300046.1425300046.1425300046.1; __utmb=...

Cabeçalho de resposta do Servidor

Chave	Valor
Resposta	HTTP/1.1 200 OK
Date	Mon, 02 Mar 2015 12:41:02 GMT
Server	Apache
X-Pingback	http://www.fabioelisio.com/blog/xmlrpc.php
Keep-Alive	timeout=5, max=500
Connection	Keep-Alive
Content-Type	text/html; charset=UTF-8
Content-Length	19.553



#Requisição

Chave	Valor
Solicitar	GET /blog/ HTTP/1.1
Accept	text/html, application/xhtml+xml, */*
Accept-Language	pt-BR
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding	gzip, deflate
Host	www.fabioelisia.com
DNT	1
Connection	Keep-Alive
Cookie	__utma=138138919.1471094785.1425300046.1425300046.1425300046.1; __utmb=...

GET – Método de requisição

/blog/ - URI da requisição – diretório raiz do servidor, sempre com “/”

HTTP/1.1 – versão do Protocolo

Host – URL requisitada



#Resposta

Chave	Valor
Resposta	HTTP/1.1 200 OK
Date	Mon, 02 Mar 2015 12:41:02 GMT
Server	Apache
X-Pingback	http://www.fabioelisio.com/blog/xmlrpc.php
Keep-Alive	timeout=5, max=500
Connection	Keep-Alive
Content-Type	text/html; charset=UTF-8
Content-Length	19.553

Status protocolo – 200 OK = sucesso
Content-type – tipo de conteúdo HTML



#Tipos de Content-type

HTTP/1.1 usa várias construções definidas para Internet Mail (RFC 822) e o Multipurpose Internet Mail Extensions (MIME), permitindo a transmissão de conteúdo em uma ampla e aberta variedade de representações

text/html	HTML	.html
text/css	Cascade Style Sheet (CSS)	.css
text/plain	Texto puro	.txt
text/xml	XML	.xml
application/javascript	JavaScript / ECMA Script	.js
application/json	JSON Document	.json, .js
application/zip	Archive ZIP	.zip
application/vnd.ms-excel	Microsoft Excel	.xls
application/mp4	Mídia MPEG-4	.mp4
image/jpeg	Imagem JPEG	.jpg, .jpeg
image/gif	Imagem GIF	.gif
image/png	Imagem PNG	.png



#Qual a diferença entre URL e URI?

Segundo a RFC 3986 (<http://www.ietf.org/rfc/rfc3986.txt>), uma URI pode ser classificada como um localizador, um nome, ou ambos. Já uma URL é um subconjunto das URIs que, além de identificar um recurso, provê meios para localizar o recurso através da descrição do mecanismo de acesso primário. Em geral, podemos identificar se um determinado caminho é uma URL se a mesma contiver o protocolo utilizado para acesso.



URI - Identificador Uniforme de Recursos

URL - Localizador de Recursos Uniforme

URN - Nome de Recursos Uniforme



#Método de requisição HTTP

- **GET** – Solicita recurso ou arquivo, dados anexados aqui são visíveis ao usuário, não é seguro para disparar ações
- **POST** – Envia dados para o servidor (ex: formulario HTML) são incluídos no corpo da requisição
- **DELETE** – Remove um recurso, retorna 204 caso não ache
- **PUT** – Atualiza recurso, pode criar caso não exista, diferença entre o post, é que o post processa informações
- **HEAD** – retorna informações de um recurso sem retorna o recurso no corpo da requisição

Outros métodos – **OPTIONS** (recupera métodos HTTP), **TRACE** (Ecoa o pedido para outros servidores), **CONNECT** (proxy para túnel SSL)



#Padrão REST

- 2000 – Dr. Roy Fielding (um dos criadores do protocolo HTTP) escreve o termo “Transferência de estado representacional” – **REST**
- Estilo de arquitetura para serviços Web
- Sistemas que seguem o principio do REST são chamados de RESTful
- Usa os diferentes métodos de comunicação (GET,POST,PUT,DELETE,HEAD e OPTIONS)
- Utiliza os *headers* do HTTP
- Definição de arquivo como recurso(cada um com seu próprio endereço)
- Utilização de *media types*
- Utilização de códigos de status HTTP
- URLs bem definidas para os recursos (ex: <http://www.xxx.com/empresas>)



Roy Fielding

Roy Fielding é um cientista de computação norte-americano. [Wikipédia](#)

Nascimento: 1965, Laguna Beach, Califórnia, EUA

Educação: Universidade da Califórnia Irvine (2000)



#Status de requisição HTTP

Toda requisição ao ser retornado tem um status. Ele vai indicar o estado da operação. As mais usadas são:

- 200 OK – A requisição foi bem sucedida
- 201 Created – Dado foi inserido com sucesso
- 202 Accepted – Pedido aceito para processamento
- 204 No Content – sem conteúdo
- 304 Not Modified – conteúdo não foi modificado
- 400 Bad Request – Requisição com problemas
- 401 Unauthorized – Usuário não autorizado
- 403 Forbidden – Sem permissão de acesso
- 404 Not Found – Pagina não encontrada
- 405 Method Not Allowed – Metodo de request não suportado
- 409 Conflict – Conflito no pedido de requisição
- 500 Internal Server Error – Erro no servidor ao processar solicitação(BUG)